Peach Frontend

# API Integrations & RPC Endpoints Documentation

Technical Documentation  
Version 1.0  
Generated from codebase analysis

# API Integrations Documentation

This document provides a comprehensive overview of all external API integrations used in the Peach Frontend application, including their endpoints, authentication methods, purposes, and configurations.

## Table of Contents

1. [Birdeye API](#birdeye-api)
2. [Mango Data API](#mango-data-api)
3. [Mango Router API](#mango-router-api)
4. [Mango Data OpenBook API](#mango-data-openbook-api)
5. [Jupiter APIs](#jupiter-apis)
6. [Notifications API](#notifications-api)
7. [Whitelist API](#whitelist-api)
8. [Dexscreener API](#dexscreener-api)
9. [OpenSerum API](#openserum-api)
10. [Pyth Network](#pyth-network)
11. [Health Check APIs](#health-check-apis)
12. [Sentry Monitoring](#sentry-monitoring)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Birdeye API

**Location:** `/apis/birdeye/helpers.ts`, `/apis/traffic/helpers.ts`, `/pages/api/tokens.ts`

### Endpoints

* \*\*Base URL\*\*: `https://public-api.birdeye.so/`
* \*\*WebSocket\*\*: `wss://public-api.birdeye.so/socket`
* \*\*Token List\*\*: `defi/tokenlist?sort\_by=v24hUSD&sort\_type=desc&offset={offset}&limit={limit}`
* \*\*Price History\*\*: `defi/history\_price?address={address}&address\_type=token&type={type}&time\_from={start}&time\_to={end}`

### Authentication

* \*\*Method\*\*: API Key
* \*\*Header\*\*: `X-API-KEY`
* \*\*Chain Header\*\*: `x-chain: solana`
* \*\*Environment Variable\*\*: `NEXT\_PUBLIC\_BIRDEYE\_API\_KEY`
* \*\*Default Fallback\*\*: `5afdc994b457493ea9a8882fbf695f46`

### Purpose/Functionality

* Token price data and historical charts
* Token list with volume filtering
* Real-time price updates via WebSocket
* Market data for trading interface

### Request/Response Formats

* \*\*Request\*\*: JSON over HTTPS/WSS
* \*\*Response\*\*: JSON format with data wrapper
* \*\*Rate Limits\*\*: Not explicitly specified in code

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Mango Data API

**Location:** `/utils/constants.ts`, `/hooks/useFollowedAccounts.ts`, `/hooks/useProfileDetails.ts`, `/apis/mngo/index.ts`

### Endpoints

* \*\*Base URL\*\*: `https://api.mngo.cloud/data/v4`
* \*\*User Following\*\*: `/user-data/following?wallet-pk={walletPk}`
* \*\*Profile Details\*\*: `/user-data/profile-details?wallet-pk={walletPk}`
* \*\*Private Accounts\*\*: `/user-data/private-accounts`
* \*\*Account Hidden\*\*: `/user-data/account-hidden?mango-account={mangoAccount}`
* \*\*Filled Orders\*\*: `/user-data/filled-orders?mango-account={mangoAccount}&id={orderIds}`
* \*\*Token Stats\*\*: `/token-historical-stats?mango-group={groupId}&mint={mint}`
* \*\*Leaderboard\*\*: `/leaderboard-season-rank?season-id={seasonId}`
* \*\*Account Tier\*\*: `/account-tier?mango-account={account}&season-id={seasonId}`
* \*\*Account Points\*\*: `/account-points-and-rank?mango-account={account}&season-id={seasonId}`
* \*\*Current Season\*\*: `/current-season`

### Authentication

* \*\*Method\*\*: No authentication required for most endpoints
* \*\*Query Parameters\*\*: wallet-pk, mango-account for user-specific data

### Purpose/Functionality

* User account data and social features
* Trading history and statistics
* Leaderboard and rewards system
* Account privacy settings
* Token performance metrics

### Request/Response Formats

* \*\*Request\*\*: GET requests with query parameters
* \*\*Response\*\*: JSON format
* \*\*Caching\*\*: 10 minutes cache time, 1 minute stale time
* \*\*Rate Limits\*\*: 3 retries on failure

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Mango Router API

**Location:** `/utils/constants.ts`

### Endpoints

* \*\*Base URL\*\*: `https://autobahn.mngo.cloud/m4b137o2h72m`

### Authentication

* \*\*Method\*\*: Not specified in available code

### Purpose/Functionality

* Route optimization for trading operations
* Transaction routing and execution

### Request/Response Formats

* \*\*Format\*\*: Not detailed in available code
* \*\*Rate Limits\*\*: Not specified

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Mango Data OpenBook API

**Location:** `/utils/constants.ts`

### Endpoints

* \*\*Base URL\*\*: `https://api.mngo.cloud/openbook/v1`

### Authentication

* \*\*Method\*\*: Not specified in available code

### Purpose/Functionality

* OpenBook market data integration
* Order book and trading data

### Request/Response Formats

* \*\*Format\*\*: Not detailed in available code
* \*\*Rate Limits\*\*: Not specified

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Jupiter APIs

**Location:** `/utils/constants.ts`

### Endpoints

* \*\*Token List Mainnet\*\*: `https://token.jup.ag/all`
* \*\*Token List Devnet\*\*: `https://api.jup.ag/api/tokens/devnet`
* \*\*Price API\*\*: `https://price.jup.ag/v4/`
* \*\*Quote API V6\*\*: `https://quote-api.jup.ag/v6`

### Authentication

* \*\*Method\*\*: No authentication required

### Purpose/Functionality

* Token metadata and listings
* Price feeds for supported tokens
* Swap quote calculations
* DEX aggregation services

### Request/Response Formats

* \*\*Request\*\*: GET requests
* \*\*Response\*\*: JSON format
* \*\*Rate Limits\*\*: Not specified in code

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Notifications API

**Location:** `/apis/notifications/`, `/utils/constants.ts`

### Endpoints

* \*\*Base URL\*\*: `https://notifications-api.herokuapp.com/`
* \*\*WebSocket\*\*: `wss://notifications-api.herokuapp.com/ws`
* \*\*Settings\*\*: `/notifications/user/getSettings`

### Authentication

* \*\*Method\*\*: Token-based authentication
* \*\*Headers\*\*:
* `authorization: {token}`
* `publickey: {walletAddress}`
* `mango-account: {mangoAccount}`

### Purpose/Functionality

* Real-time notifications for trading events
* User notification preferences
* WebSocket connection for live updates
* Fill notifications and alerts

### Request/Response Formats

* \*\*Request\*\*: JSON over HTTPS/WSS
* \*\*Response\*\*: JSON format
* \*\*WebSocket\*\*: Ping/pong every 30 seconds
* \*\*Rate Limits\*\*: 2 retry attempts with 5-second delay

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Whitelist API

**Location:** `/apis/whitelist.ts`, `/utils/constants.ts`

### Endpoints

* \*\*Base URL\*\*: `https://api.mngo.cloud/whitelist/v1/`
* \*\*Check Whitelist\*\*: `/isWhiteListed?wallet={wallet}`

### Authentication

* \*\*Method\*\*: No authentication required
* \*\*Query Parameter\*\*: wallet address

### Purpose/Functionality

* Wallet whitelist verification
* Access control for restricted features

### Request/Response Formats

* \*\*Request\*\*: GET with query parameters
* \*\*Response\*\*: JSON with `found` boolean field
* \*\*Rate Limits\*\*: Not specified

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Dexscreener API

**Location:** `/components/governance/ListToken/ListToken.tsx`

### Endpoints

* \*\*Base URL\*\*: `https://api.dexscreener.com/latest/dex/search`
* \*\*Token Search\*\*: `/search?q={tokenMint}`

### Authentication

* \*\*Method\*\*: No authentication required

### Purpose/Functionality

* Token pair discovery
* Pool information for Raydium and Orca
* Market data for governance token listings

### Request/Response Formats

* \*\*Request\*\*: GET with query parameters
* \*\*Response\*\*: JSON with pairs array
* \*\*Rate Limits\*\*: Not specified

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## OpenSerum API

**Location:** `/next.config.js`

### Endpoints

* \*\*Proxied URL\*\*: `/openSerumApi/:path\*`
* \*\*Actual URL\*\*: `https://openserum.io/api/serum/:path\*`

### Authentication

* \*\*Method\*\*: Proxied through Next.js (CORS bypass)

### Purpose/Functionality

* Serum DEX market data
* Order book information
* Trading pair details

### Request/Response Formats

* \*\*Request\*\*: Proxied through Next.js rewrites
* \*\*Response\*\*: JSON format
* \*\*Rate Limits\*\*: Not specified

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Pyth Network

**Location:** `/utils/governance/listingTools.ts`, `/pages/dashboard/index.tsx`

### Endpoints

* \*\*Program ID\*\*: `MAINNET\_PYTH\_PROGRAM`
* \*\*HTTP Client\*\*: `PythHttpClient`

### Authentication

* \*\*Method\*\*: On-chain program calls

### Purpose/Functionality

* Oracle price feeds
* Real-time price data for tokens
* Price validation for governance proposals

### Request/Response Formats

* \*\*Request\*\*: Solana program calls
* \*\*Response\*\*: On-chain data structures
* \*\*Rate Limits\*\*: Blockchain network limits

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Health Check APIs

**Location:** `/hooks/useOffchainServicesHealth.ts`

### Endpoints

* \*\*Database Health\*\*: `https://api.mngo.cloud/data/health/db`
* \*\*Redis Health\*\*: `https://api.mngo.cloud/data/health/redis`
* \*\*Server Health\*\*: `https://api.mngo.cloud/data/health/server`

### Authentication

* \*\*Method\*\*: No authentication required

### Purpose/Functionality

* System health monitoring
* Service availability checks
* Status page integration

### Request/Response Formats

* \*\*Request\*\*: GET requests
* \*\*Response\*\*: HTTP status codes (200, 300, 500)
* \*\*Polling\*\*: Every 60 seconds
* \*\*Rate Limits\*\*: Not specified

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Sentry Monitoring

**Location:** `/next.config.js`

### Configuration

* \*\*Organization\*\*: `mango`
* \*\*Project\*\*: `mango-v4-ui`
* \*\*Tunnel Route\*\*: `/monitoring`

### Authentication

* \*\*Method\*\*: Sentry DSN configuration

### Purpose/Functionality

* Error tracking and monitoring
* Performance monitoring
* Source map uploads
* Client-side error reporting

### Request/Response Formats

* \*\*Request\*\*: Automated error reports
* \*\*Response\*\*: Sentry platform processing
* \*\*Rate Limits\*\*: Sentry platform limits

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Summary

This application integrates with **12 major external APIs** and services, ranging from market data providers to monitoring solutions. Most APIs use simple HTTP GET requests with JSON responses, while some require API keys or token-based authentication. The application implements proper error handling, retry mechanisms, and caching strategies for optimal performance and reliability.

**Key Integration Patterns:**

* REST APIs with JSON responses
* WebSocket connections for real-time data
* API key authentication where required
* Proxy configurations for CORS handling
* Health monitoring and error tracking
* Caching and retry strategies for reliability

# RPC Endpoints Documentation

This document provides a comprehensive overview of all RPC (Remote Procedure Call) endpoints and configurations used in the Peach Frontend application, including protocols, methods, connection details, and error handling mechanisms.

## Table of Contents

1. [Solana RPC Endpoints](#solana-rpc-endpoints)
2. [Lite RPC Configuration](#lite-rpc-configuration)
3. [WebSocket RPC Connections](#websocket-rpc-connections)
4. [Mango v4 Program RPC](#mango-v4-program-rpc)
5. [Jupiter RPC Integration](#jupiter-rpc-integration)
6. [Governance RPC Calls](#governance-rpc-calls)
7. [Error Handling Mechanisms](#error-handling-mechanisms)
8. [Connection Management](#connection-management)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Solana RPC Endpoints

**Location:** `/components/settings/RpcSettings.tsx`, `/store/mangoStore.ts`, `/utils/constants.ts`

### Protocol

* \*\*Type\*\*: JSON-RPC 2.0 over HTTPS/WSS
* \*\*Transport\*\*: HTTP/HTTPS for standard calls, WebSocket for subscriptions

### Available RPC Providers

#### Mainnet Providers

1. \*\*Triton (Recommended)\*\*

* \*\*URL\*\*: `https://mango.rpcpool.com/{API\_KEY}`
* \*\*WebSocket\*\*: `wss://mango.rpcpool.com/{API\_KEY}`
* \*\*Authentication\*\*: API Key required
* \*\*Features\*\*: High performance, low latency

1. \*\*Helius\*\*

* \*\*URL\*\*: `https://mainnet.helius-rpc.com/?api-key={API\_KEY}`
* \*\*WebSocket\*\*: `wss://mainnet.helius-rpc.com/?api-key={API\_KEY}`
* \*\*Authentication\*\*: API Key required
* \*\*Features\*\*: Enhanced RPC methods

1. \*\*QuickNode\*\*

* \*\*URL\*\*: `https://example.solana-mainnet.quiknode.pro/{API\_KEY}/`
* \*\*WebSocket\*\*: `wss://example.solana-mainnet.quiknode.pro/{API\_KEY}/`
* \*\*Authentication\*\*: API Key required
* \*\*Features\*\*: Global infrastructure

1. \*\*Solana Foundation\*\*

* \*\*URL\*\*: `https://api.mainnet-beta.solana.com`
* \*\*WebSocket\*\*: `wss://api.mainnet-beta.solana.com`
* \*\*Authentication\*\*: None (public)
* \*\*Features\*\*: Official Solana RPC

#### Devnet Providers

1. \*\*Solana Devnet\*\*

* \*\*URL\*\*: `https://api.devnet.solana.com`
* \*\*WebSocket\*\*: `wss://api.devnet.solana.com`
* \*\*Authentication\*\*: None (public)
* \*\*Features\*\*: Development testing

### RPC Methods Used

#### Account Operations

* `getAccountInfo`: Retrieve account data
* `getMultipleAccounts`: Batch account retrieval
* `getProgramAccounts`: Filter accounts by program

#### Transaction Operations

* `sendTransaction`: Submit transactions
* `confirmTransaction`: Transaction confirmation
* `getTransaction`: Retrieve transaction details
* `getSignatureStatuses`: Check transaction status
* `simulateTransaction`: Dry-run transactions

#### Block and Slot Operations

* `getSlot`: Current slot number
* `getBlockHeight`: Current block height
* `getRecentBlockhash`: Recent blockhash for transactions
* `getFeeForMessage`: Calculate transaction fees

#### Subscription Methods (WebSocket)

* `accountSubscribe`: Account change notifications
* `programSubscribe`: Program account notifications
* `signatureSubscribe`: Transaction confirmation
* `slotSubscribe`: Slot change notifications
* `blockPrioritizationFeesSubscribe`: Fee estimation

### Input/Output Schemas

#### Standard Request Format

{  
 "jsonrpc": "2.0",  
 "id": 1,  
 "method": "methodName",  
 "params": []  
}

#### Standard Response Format

{  
 "jsonrpc": "2.0",  
 "id": 1,  
 "result": {},  
 "error": {  
 "code": -32000,  
 "message": "Error description"  
 }  
}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Lite RPC Configuration

**Location:** `/components/MangoProvider.tsx`, `/utils/constants.ts`

### Protocol

* \*\*Type\*\*: JSON-RPC 2.0 over WebSocket
* \*\*Purpose\*\*: Fee estimation and priority fee subscriptions

### Endpoints

* \*\*Mainnet\*\*: `wss://mango.rpcpool.com/{API\_KEY}` (converted from HTTPS)
* \*\*Connection\*\*: Automatic HTTPS to WSS conversion

### Methods

* `blockPrioritizationFeesSubscribe`: Real-time fee estimation
* \*\*Parameters\*\*:
* `addresses`: Array of account addresses to monitor
* `commitment`: Confirmation level ("processed", "confirmed", "finalized")

### Response Format

{  
 "subscription": 123,  
 "result": {  
 "prioritizationFee": 5000,  
 "slot": 123456789  
 }  
}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## WebSocket RPC Connections

**Location:** `/components/MangoProvider.tsx`, `/hooks/useConnection.ts`

### Connection Management

* \*\*Auto-reconnection\*\*: Implemented with exponential backoff
* \*\*Heartbeat\*\*: Ping/pong mechanism for connection health
* \*\*Subscription Management\*\*: Automatic re-subscription on reconnect

### Subscription Types

#### Account Subscriptions

* \*\*Method\*\*: `accountSubscribe`
* \*\*Purpose\*\*: Monitor account balance and data changes
* \*\*Parameters\*\*: Account public key, commitment level

#### Program Subscriptions

* \*\*Method\*\*: `programSubscribe`
* \*\*Purpose\*\*: Monitor all accounts owned by a program
* \*\*Parameters\*\*: Program ID, filters, commitment level

#### Signature Subscriptions

* \*\*Method\*\*: `signatureSubscribe`
* \*\*Purpose\*\*: Transaction confirmation notifications
* \*\*Parameters\*\*: Transaction signature, commitment level

### Error Handling

* \*\*Connection Errors\*\*: Automatic retry with backoff
* \*\*Subscription Errors\*\*: Re-subscription attempts
* \*\*Timeout Handling\*\*: 30-second timeout for responses

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Mango v4 Program RPC

**Location:** `/utils/mango.ts`, `/store/mangoStore.ts`, `/pages/api/tokentiers.ts`

### Protocol

* \*\*Type\*\*: Solana Program RPC calls
* \*\*Program ID\*\*: `4MangoMjqJ2firMokCjjGgoK8d4MXcrgL7XJaL3w6fVg`

### Available Methods

#### Account Management

* `getMangoAccount`: Retrieve Mango account data
* `getMangoGroup`: Get group configuration
* `getAllMangoAccounts`: Batch account retrieval

#### Market Operations

* `getPerpMarkets`: Perpetual market data
* `getSpotMarkets`: Spot market information
* `getTokens`: Token configurations

#### Trading Operations

* `placePerpOrder`: Place perpetual orders
* `placeSpotOrder`: Place spot orders
* `cancelOrder`: Cancel existing orders
* `settleFunds`: Settle trading funds

### Input Parameters

* \*\*Connection\*\*: Solana RPC connection object
* \*\*Group\*\*: Mango group public key
* \*\*Account\*\*: User's Mango account public key
* \*\*Market\*\*: Market public key for trading operations

### Response Formats

* \*\*Account Data\*\*: Deserialized account structures
* \*\*Market Data\*\*: Market configuration and state
* \*\*Transaction Results\*\*: Transaction signatures and confirmations

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Jupiter RPC Integration

**Location:** `/utils/jupiter.ts`, `/components/swap/`

### Protocol

* \*\*Type\*\*: HTTP REST API (not traditional RPC)
* \*\*Purpose\*\*: DEX aggregation and swap routing

### Endpoints

* \*\*Quote API\*\*: `https://quote-api.jup.ag/v6/quote`
* \*\*Swap API\*\*: `https://quote-api.jup.ag/v6/swap`
* \*\*Price API\*\*: `https://price.jup.ag/v4/price`

### Methods

#### Quote Generation

* \*\*Endpoint\*\*: `/quote`
* \*\*Parameters\*\*:
* `inputMint`: Source token mint
* `outputMint`: Destination token mint
* `amount`: Swap amount
* `slippageBps`: Slippage tolerance

#### Swap Execution

* \*\*Endpoint\*\*: `/swap`
* \*\*Parameters\*\*: Quote object, user public key
* \*\*Response\*\*: Serialized transaction for signing

### Error Handling

* \*\*Rate Limiting\*\*: Automatic retry with exponential backoff
* \*\*Invalid Routes\*\*: Fallback to alternative routes
* \*\*Network Errors\*\*: Connection retry mechanism

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Governance RPC Calls

**Location:** `/utils/governance/tools.ts`, `/utils/governance/listingTools.ts`

### Protocol

* \*\*Type\*\*: Solana Program RPC calls
* \*\*Program ID\*\*: `GovER5Lthms3bLBqWub97yVrMmEogzX7xNjdXpPPCVZw`

### Available Methods

#### Realm Operations

* `fetchRealm`: Get governance realm data
* `fetchGovernances`: Retrieve governance accounts
* `fetchProposals`: Get active proposals

#### Voting Operations

* `castVote`: Submit governance votes
* `createProposal`: Create new proposals
* `executeInstruction`: Execute approved instructions

### Input Parameters

* \*\*Connection\*\*: Solana RPC connection
* \*\*Realm\*\*: Governance realm public key
* \*\*Governance\*\*: Specific governance public key
* \*\*Proposal\*\*: Proposal public key for voting

### Response Formats

* \*\*Realm Data\*\*: Governance configuration and state
* \*\*Proposals\*\*: Proposal details and voting status
* \*\*Vote Records\*\*: Individual vote information

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Error Handling Mechanisms

### Connection Errors

#### Network Failures

* \*\*Retry Strategy\*\*: Exponential backoff (1s, 2s, 4s, 8s)
* \*\*Max Retries\*\*: 5 attempts
* \*\*Fallback\*\*: Switch to alternative RPC provider

#### Timeout Handling

* \*\*Request Timeout\*\*: 30 seconds
* \*\*WebSocket Timeout\*\*: 60 seconds
* \*\*Subscription Timeout\*\*: 120 seconds

### RPC-Specific Errors

#### Rate Limiting (HTTP 429)

* \*\*Response\*\*: Automatic retry with increased delay
* \*\*Backoff\*\*: Linear increase (5s, 10s, 15s)
* \*\*Provider Switch\*\*: Rotate to different RPC endpoint

#### Invalid Parameters (HTTP 400)

* \*\*Response\*\*: Log error and return user-friendly message
* \*\*Validation\*\*: Client-side parameter validation
* \*\*Recovery\*\*: Prompt user to correct input

#### Server Errors (HTTP 5xx)

* \*\*Response\*\*: Retry with exponential backoff
* \*\*Fallback\*\*: Switch to backup RPC provider
* \*\*Monitoring\*\*: Error reporting to Sentry

### Transaction Errors

#### Insufficient Funds

* \*\*Detection\*\*: Pre-flight simulation
* \*\*Response\*\*: User notification with balance check
* \*\*Recovery\*\*: Suggest funding account

#### Blockhash Expiration

* \*\*Detection\*\*: Transaction confirmation timeout
* \*\*Response\*\*: Automatic retry with fresh blockhash
* \*\*Prevention\*\*: Recent blockhash caching

#### Network Congestion

* \*\*Detection\*\*: High priority fees required
* \*\*Response\*\*: Dynamic fee adjustment
* \*\*User Control\*\*: Fee preference settings

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Connection Management

### Connection Pooling

**Location:** `/store/mangoStore.ts`

#### Pool Configuration

* \*\*Max Connections\*\*: 10 per RPC endpoint
* \*\*Connection Reuse\*\*: HTTP/1.1 keep-alive
* \*\*Load Balancing\*\*: Round-robin across providers

#### Health Monitoring

* \*\*Ping Interval\*\*: 30 seconds
* \*\*Health Check\*\*: `getSlot` method call
* \*\*Failure Threshold\*\*: 3 consecutive failures

### Provider Switching

#### Automatic Failover

* \*\*Trigger\*\*: Connection failure or high latency
* \*\*Selection\*\*: Next available provider in priority order
* \*\*Notification\*\*: User alert for manual provider changes

#### Manual Selection

* \*\*Interface\*\*: RPC Settings component
* \*\*Persistence\*\*: Local storage for user preference
* \*\*Validation\*\*: Connection test before switching

### Performance Optimization

#### Request Batching

* \*\*Method\*\*: `getMultipleAccounts` for batch operations
* \*\*Batch Size\*\*: Maximum 100 accounts per request
* \*\*Parallel Requests\*\*: Up to 5 concurrent batches

#### Caching Strategy

* \*\*Account Data\*\*: 30-second cache for static data
* \*\*Market Data\*\*: 5-second cache for dynamic data
* \*\*Blockhash\*\*: 60-second cache with refresh

#### Subscription Management

* \*\*Deduplication\*\*: Single subscription per unique filter
* \*\*Cleanup\*\*: Automatic unsubscribe on component unmount
* \*\*Reconnection\*\*: Restore all subscriptions after reconnect

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Summary

This application utilizes **multiple RPC protocols and endpoints** for comprehensive blockchain interaction:

* \*\*Primary Protocol\*\*: Solana JSON-RPC 2.0
* \*\*Transport Methods\*\*: HTTPS, WebSocket
* \*\*Provider Options\*\*: 5 mainnet, 1 devnet
* \*\*Program Integrations\*\*: Mango v4, Jupiter, Governance
* \*\*Error Handling\*\*: Comprehensive retry and failover mechanisms
* \*\*Performance Features\*\*: Connection pooling, caching, batching

**Key RPC Patterns:**

* JSON-RPC 2.0 standard compliance
* WebSocket subscriptions for real-time data
* Automatic provider failover and health monitoring
* Comprehensive error handling and retry logic
* Performance optimization through caching and batching
* User-configurable RPC provider selection